Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Cancelled)
2. (Cancelled)
3. (Cancelled)
4. (Cancelled)

5. (previously presented) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

determining whether an operation request is illegal or harmful to an environment of an application, and

preventing said application from executing an illegal or harmful operation request, wherein said step of preventing comprises the step of modifying said illegal or harmful operation request into a legal or harmless operation request.

6. (previously presented) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

determining whether an operation request is illegal or harmful to an environment of an application, and

preventing said application from executing an illegal or harmful operation request, wherein said step of preventing comprises the step of replacing said illegal or harmful operation request with a legal or harmless operation request.

7. (previously presented) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

designating an application path of an application as restricted,

determining whether an operation request is illegal or harmful to an environment of said application, and

preventing said application from executing an illegal or harmful operation request,

wherein said step of determining comprises the step of checking said operation request for an existence of an embedded command causing database manipulation.

8. (previously presented) The method of claim 7, wherein said embedded command is an SQL based command.

9. (original) The method of claim 7, further comprising the steps of:

parsing said operation request into one or more expressions;

building a state-automate;

inspecting said one or more expressions for improper syntax and characters not defined in a first alphabet; and

applying said state-automate to said operation request.

10. (original) The method of claim 9, wherein said alphabet is selected from the group consisting of: letters, digits, and encoded characters; blocks of letters; groups of said blocks; and any combination thereof.

11. (Cancelled)

12. (Cancelled)

13. (previously presented) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

determining whether an operation request is illegal or harmful to an environment of an application,

preventing said application from executing an illegal or harmful operation request,

comparing said operation request against stored known vulnerability patterns to determine a match, and

blocking said operation request if said match is found,

wherein said step of comparing comprises the steps of:

converting every consecutive specified number of characters in said operation request into n-bits of binary code;

computing a hash value for said every consecutive specified number of characters in said operation request; and

comparing every hash value to stored hash values representing vulnerability patterns.

14. (original) The method of claim 13, wherein said n-bits is 8 bits and said specified number is equal to four.

15. (Cancelled)

16. (Cancelled)

17. (previously presented) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

determining whether an operation request is illegal or harmful to an environment of an application;

preventing said application from executing an illegal or harmful operation request;

sending a legal or harmless operation requests to said application; and

generating a reply to said operation request.

18. (original) The method of claim 17, wherein said step of determining further comprises the steps of:

determining a first set of internal URLs and parameters values contained in said reply;

receiving a second operation request in response to said reply;

comparing a second set of internal URLs and parameters values contained in said second operation request with said first set to determine if said sets correspond; and

rejecting said second operation request if said sets do not correspond.

19. (original) The method of claim 17, further comprising the steps of:

identifying a single client to interact with said application;

determining a first set of parameter names and values in said reply;

receiving a second operation request from said client in response to said reply;

determining a second set of parameter names and values in said second operation request; and

forwarding said second request to said application only if said first set matches said second set.

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (currently amended) A method for protecting an application from executing an illegal or harmful operation request received from a distrusted environment, the method comprising the steps of:

determining whether an operation request is illegal or harmful to an environment of an application, and

preventing said application from executing an illegal or harmful operation request, wherein said step of determining comprises the steps of:

identifying a cookie message header in said operation request;

decrypting values in said cookie message header; and

~~modifyng~~ modifying said operation request to reflect said decrypted values.

24. (original) The method of claim 17, further comprising the steps of:

identifying a cookie message header in said reply;

encrypting values in said cookie message header; and

modifying said reply to reflect said encrypted values.

25. (Cancelled)

26. (Cancelled)

27. (Cancelled)

28. (Cancelled)

29. (Cancelled)

30. (currently amended) A method for preventing one or more applications from executing out of their intended scopes of operation, comprising the steps of:

receiving one or more operation requests;

formatting each operation request into a formatted message according to a designated communications protocol, wherein said designation communication protocol is determined by the type of application being requested;

indexing said one or more formatted messages;

storing a copy of said indexed one or more formatted messages;

translating said formatted messages into internal messages according to an encoding scheme;

resolving a destination node for each operation request;

matching each operation request to an application path, wherein said application path is a virtual directory or a subdirectory of said application; and

determining whether each operation request is illegal or harmful to an environment of said application, wherein said step of determining comprises the step of:

applying one or more security pipes to each operation request, wherein the number and types of pipes applied to each operation request are based on said resolved destination node of each operation request ~~The method of claim 27~~, wherein application of a pipe comprises the steps of:

parsing a first operation request into one or more expressions;

building a state-automate;

inspecting said one or more expressions for improper syntax and characters not defined in a first alphabet; and

applying said state-automate to said first operation request.

31. (currently amended) The method of claim 30, wherein said alphabet is selected from the group consisting of: letters, digits, and encoded characters; blocks of letters; groups of said blocks; and any combination thereof.

32. (Cancelled)

33. (Cancelled)

34. (currently amended) A method for preventing one or more applications from executing out of their intended scopes of operation, comprising the steps of:

receiving one or more operation requests;

formatting each operation request into a formatted message according to a designated communications protocol, wherein said designation communication protocol is determined by the type of application being requested;

indexing said one or more formatted messages;

storing a copy of said indexed one or more formatted messages;

translating said formatted messages into internal messages according to an encoding scheme;

resolving a destination node for each operation request;

matching each operation request to an application path, wherein said application path is a virtual directory or a subdirectory of said application; and

determining whether each operation request is illegal or harmful to an environment of said application, wherein said step of determining comprises the step of:

applying one or more security pipes to each operation request, wherein the number and types of pipes applied to each operation request are based on said resolved destination node of each operation request, wherein application of a pipe comprises the steps of:

comparing said operation request against stored known vulnerability patterns to determine a match, ~~The method of claim 27~~, wherein said step of comparing comprises the steps of:

converting every consecutive specified number of characters in said operation request into n-bits of binary code;

computing a hash value for said every consecutive specified number of characters in said operation request; and

comparing every hash value to stored hash values representing vulnerability patterns; and

blocking said operation request if said match is found.

35. (original) The method of claim 34, wherein said n-bits is 8-bits and said specified number is equal to four.

36. (Cancelled)

37. (Cancelled)

38. (currently amended) A method for preventing one or more applications from executing out of their intended scopes of operation, comprising the steps of:

receiving one or more operation requests;

formatting each operation request into a formatted message according to a designated communications protocol, wherein said designation communication protocol is determined by the type of application being requested;

indexing said one or more formatted messages;

storing a copy of said indexed one or more formatted messages;

translating said formatted messages into internal messages according to an encoding scheme;

resolving a destination node for each operation request;

matching each operation request to an application path, wherein said application path is a virtual directory or a subdirectory of said application;

determining whether each operation request is illegal or harmful to an environment of said application, wherein said step of determining comprises the step of:

applying one or more security pipes to each operation request, wherein the number and types of pipes applied to each operation request are based on said resolved destination node of each operation request;

~~The method of claim 27, further comprising the steps of:~~

sending legal or harmless operation requests to said operation; and

generating a reply to said operation request.

39. (original) The method of claim 38, wherein application of a pipe comprises the steps of:

determining a first set of internal URLs and parameters values contained in said reply;

receiving a second operation request in response to said reply;

comparing a second set of internal URLs and parameters values contained in said second operation request with said first set to determine if said sets correspond; and

rejecting said second operation request if said sets do not correspond.

40. (original) The method of claim 38, wherein application of a pipe comprises the steps of:

identifying a single client to interact with said application;

determining a first set of parameter names and values in said reply;

receiving a second operation request from said client in response to said reply;

determining a second set of parameter names and values in said second operation request; and

forwarding said second request to said application only if said first set matches said second set.

41. (Cancelled)

42. (Cancelled)

43. (Cancelled)

44. (previously presented)A method for preventing one or more applications from executing out of their intended scopes of operation, comprising the steps of:

receiving one or more operation requests;

formatting each operation request into a formatted message according to a designated communications protocol, wherein said designation communication protocol is determined by the type of application being requested;

indexing said one or more formatted messages;

storing a copy of said indexed one or more formatted messages;

translating said formatted messages into internal messages according to an encoding scheme;

resolving a destination node for each operation request; and

applying one or more security pipes to each operation request, wherein the number and types of pipes applied to each operation request are based on said resolved destination node of each operation request, and wherein application of a pipe comprises the steps of:

identifying a cookie message header in said operation request;

decrypting values in said cookie message header; and

modifying said operation request to reflect said decrypted values.

45. (previously presented) A method for preventing one or more applications from executing out of their intended scopes of operation, comprising the steps of:

receiving one or more operation requests;

formatting each operation request into a formatted message according to a designated communications protocol, wherein said designation communication protocol is determined by the type of application being requested;

indexing said one or more formatted messages;

storing a copy of said indexed one or more formatted messages;

translating said formatted messages into internal messages according to an encoding scheme;

resolving a destination node for each operation request; and

applying one or more security pipes to each operation request, wherein the number and types of pipes applied to each operation request are based on said resolved destination node of each operation request, and wherein application of a pipe comprises the steps of:

identifying a cookie message header in said reply;

encrypting values in said cookie message header; and

modifying said reply to reflect said encrypted values.

46. (Cancelled)

47. (Cancelled)

48. (Cancelled)

49. (Cancelled)

50. (Cancelled)

51. (Cancelled)

52. (Cancelled)

53. (Cancelled)

54. (Cancelled)